2024 - 2025

ΜΥΕΟ3Ο ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Εργασία του Ντούτση Παναγιώτη Δαυίδ **(5479)**

Περιεχόμενα

0. Abstract

Η εφαρμογή συνιστά μια ολοκληρωμένη λύση διαχείρισης, επεξεργασίας και οπτικοποίησης δεδομένων προερχόμενα από csv αρχεία, με τελικό «αποδέκτη» μια βάση δεδομένων PostgreSQL και διεπαφή χρήστη υλοποιημένη με Javascript stack (node, express & react).

Πιο αναλυτικά, αναπτύχθηκε ένας μηχανισμός ETL (Extract–Transform–Load), με τη γλώσσα python, ο οποίος αναγνωρίζει, “καθαρίζει” και ενοποιεί δεδομένα από πολλαπλές πηγές, τα μετασχηματίζει και τα φορτώνει σε πίνακες της βάσης. Εν συνεχεία, το backend τροφοδοτεί το frontend μέσω RESTful API, παρέχοντας στον αναλυτή δυνατότητες φιλτραρίσματος, πλοήγησης και απεικόνισης με τη χρήση γραφημάτων και πινάκων.

Κατόπιν δοκιμών, έχει επιβεβαιωθεί η αξιοπιστία και η υψηλή ταχύτητα απόκρισης σε αιτήματα της εφαρμογής. Το περιβάλλον χρήστη είναι εύκολο στην κατανόηση, επιτρέπει πολλαπλούς τρόπους φιλτραρίσματος των δεδομένων και, ως εκ τούτου, την εύκολη μελέτη τους από τους ενδιαφερομένους

1. Σκοπός

Η ανάπτυξη της εφαρμογής επιδιώκει να καλύψει τις ανάγκες ενός αναλυτή ποδοσφαιρικών αγώνων για συνέπεια των δεδομένων και ευκολία στη χρήση, παρέχοντας αξιόπιστες αναλύσεις και ένα πλήρως αυτοματοποιημένο μηχανισμό φιλτραρίσματος και φόρτωσης σε βάση PostgreSQL δεδομένων. Ταυτόχρονα, μέσω δυναμικών queries, δίνονται πολλαπλές δυνατότητες στους αναλυτές για μελέτη των δεδομένων με χρονολογικές και γεωγραφικές

2. Tech Stack

Η Python πραγματοποιεί τον καθαρισμό των CSV σε batch, αξιοποιώντας τις δυνατότητες της PostgreSQL για γρήγορο bulk loading και indexing. Στη συνέχεια, το Express/Node.js ανοίγει REST endpoints που εκμεταλλεύονται το non-blocking I/O για άμεσες απαντήσεις. Το React καλεί αυτά τα endpoints, παρουσιάζοντας γραφήματα με component-based δομή. Η κοινή χρήση JavaScript σε backend και frontend μειώνει την ανάγκη γνώσεις πολλών γλωσσών προγραμματισμού, με συνέπεια να είναι πιο εύκολη η κατανόηση και, ενδεχομένως η περαιτέρω ανάπτυξη της εφαρμογής μελλοντικά, ενώ η Python αναλαμβάνει τις σύνθετες εργασίες ETL όπου είναι απαραίτητη. Έτσι, το stack προσφέρει ευελιξία, αποδοτικότητα και επεκτασιμότητα

3. Προαπαιτούμενα και Οδηγίες Εκτέλεσης

- **PostgreSQL** (>= 15) με δικαιώματα δημιουργίας βάσεων και χρηστών.
- **Python** (>= 3.8) και **pip**
- **Node.js** (>= 14) και **npm** ή **yarn**

Ρύθμιση Βάσης Δεδομένων:

Δημιουργία βάσης	psql -U postgres CREATE DATABASE mye030; \q
Εκτέλεση DDL	cd Backend/Database/ psql -U postgres -d mye030 -f schema.sql

Backend

Στον κώδικα υπάρχει ήδη .env αρχείο. Απλώς εκτελέστε npm install

ETL Pipeline (Python)

Εντός του φακέλου Database στον Backend, δημιουργήστε ένα εικονικό περιβάλλον	python3 -m venv venv source venv/bin/activate
Εγκαταστήστε απαιτήσεις	pip install -r requirements.txt
Τρέξτε το script	Python filter.py

Frontend

Μεταβείτε στο Frontend	npm install
Εκκίνηση vite	npm run dev

4. Υλοποίηση

ETL pipeline: Η διαδικασία ξεκινά με απλά CSV αρχεία δεδομένων και Python scripts: κάθε CSV (countries, results, goalscorers, shootouts, former_names) φορτώνεται σε pandas DataFrame, καθαρίζεται (dropna, drop_duplicates), τυποποιούνται τύποι (π.χ. ημερομηνίες, ακέραιοι) και ομαδοποιούνται κανόνες μετασχηματισμού (regex, default τιμές). Στη συνέχεια, η συνάρτηση truncate_all κάνει TRUNCATE & RESTART IDENTITY, ενώ με bulk-insert (psycopg2 + execute_values) τα προκαθαρισμένα δεδομένα φορτώνονται στα αντίστοιχα tables. Μια πρόσθετη φάση mapping ανακτά τα IDs των matches (ημερομηνία, home, away) ώστε goals και shootouts να αντιστοιχηθούν σωστά, και τυχόν ασυμφωνίες γράφονται σε quarantined αρχεία για έλεγχο

Schema & Views: Στην PostgreSQL, το schema αποτυπώνει τα entities: countries, matches, goals, penalty_shootouts και former_names. Δείκτες σε match_date, team_ids και scorer επιταχύνουν αναζητήσεις. Πάνω σ' αυτό στήνονται views που υλοποιούν τα aggregates και τα joins:

- match_details, goal_details συνδυάζουν τα raw δεδομένα με τα ονόματα των χωρών,
- summary views (tournament_summary, country_performance κ.ά.) υπολογίζουν counts, sums, averages με GROUP BY και CASE expressions,
- profile views (country_profile) ενσωματώνουν performance, activity span και home/away στατιστικά σε ένα ενιαίο αποτέλεσμα.

Backend (Node & Express): δημιουργεί RESTful endpoints που αντιστοιχούν στα views. Κάθε αίτημα περνά από τον κοινό χειριστή handleViewRequest, ο οποίος:

1. Διαβάζει τα query params,
2. Κατασκευάζει δυναμικά το WHERE (με parameterized queries) και ORDER BY,
3. Εκτελεί δύο queries (data με LIMIT/OFFSET και COUNT για pagination) και
4. Επιστρέφει το JSON { data, pagination }.

Συναρτήσεις για δυναμικά queries (head-to-head, top-countries, goal-timing) υλοποιούνται σε ξεχωριστά endpoints με custom filters και επιπλέον aggregations.

Frontend: το React App φορτώνει επιμέρους components (MatchFinder, CountryWdlChart, PlayerGoalSearch κ.ά.). Κάθε component:

- Κάνει fetch στο αντίστοιχο /api/... με φίλτρα και pagination,
- Φυλάει data, isLoading, error, pagination σε useState,
- Απεικονίζει αποτελέσματα μέσω πινάκων (react-table) ή γραφημάτων (hooks + Recharts/Chart.js).

5. Αναλυτική Περιγραφή

ETL

Μέθοδος	Περιγραφή
truncate_all(conn_params)	Κάνει TRUNCATE σε όλους τους πίνακες με RESTART IDENTITY και CASCADE.
clean_countries(path)	Φορτώνει το CSV των χωρών, απορρίπτει null και διπλότυπες εγγραφές, συμπληρώνει default τιμές, τυποποιεί στήλες και επιστρέφει pandas DataFrame.
clean_former_names(path)	Φορτώνει το CSV με πρώην ονόματα, μετονομάζει στήλες, μετατρέπει σε datetime, απορρίπτει null εγγραφές και επιστρέφει DataFrame.
clean_shootouts(path)	Φορτώνει το CSV των πέναλτι, μετατρέπει την ημερομηνία, απορρίπτει ελλείψεις σε κρίσιμα πεδία και επιστρέφει DataFrame.
clean_goalscorers(path)	Φορτώνει το CSV των σκόρερ, μετατρέπει date και minute, φιλτράρει null, μετατρέπει τα own_goal/penalty σε boolean, αφαιρεί διπλότυπα και επιστρέφει DataFrame.
clean_results(path)	Φορτώνει το CSV με αποτελέσματα αγώνων, μετατρέπει date, scores και neutral σε σωστούς τύπους, φιλτράρει null και επιστρέφει DataFrame.
add_virtual_countries(countries_df, *team_sets)*	Εντοπίζει ομάδες που λείπουν από το αρχικό countries_df και προσθέτει “virtual” εγγραφές με default πεδία.
check_consistency(countries_df, results_df, goalscorers_df, shootouts_df, former_names_df)	Ελέγχει ότι όλα τα ονόματα ομάδων/χωρών που εμφανίζονται σε δεδομένα υπάρχουν στο countries_df ή θα προστεθούν ως virtual.
insert_to_postgres(df, table_name, conn_params, allowed_columns=None)	Κάνει bulk insert ενός DataFrame σε πίνακα της PostgreSQL, μετατρέποντας τύπους και διαχειριζόμενο NULLs, με logging σε περίπτωση σφαλμάτων.
get_match_id_map(conn_params)	Ανακτά από τη βάση έναν χάρτη (match_date, home_team, away_team) → match_id για να γίνει σωστό mapping των goals και shootouts.
report_virtual_countries(conn_params)	Κάνει query στον πίνακα countries για να εμφανίσει τις εγγραφές με status ‘Unrecognized’ ή developed_or_developing ‘Unknown’.

Schema

Στην βάση υπάρχουν 5 πίνακες

1) **countries**
   Αποθηκεύει τα σταθερά μεταδεδομένα κάθε κράτους. Το primary key είναι το country_id (SERIAL), ενώ τα display_name και iso/iso3 είναι μοναδικά (UNIQUE) για γρήγορη αναφορά. Τα iso_code, fips, region_code κ.ά. είναι απλοί VARCHAR ή INT πεδία για ταξινόμηση και φιλτράρισμα. Boolean flags (sids, lldc, ldc) υποδεικνύουν ειδικές κατηγορίες χωρών. Το area_sq_km και το population είναι INT/BIGINT για μετρήσεις μεγέθους και πληθυσμού. Οι δείκτες σε display_name και iso επιταχύνουν αναζητήσεις κατά χώρα.

2) **matches**
   Κάθε γραμμή αντιστοιχεί σε έναν αγώνα: το match_id (SERIAL) είναι το primary key, και το match_date (DATE) συνοδεύεται από UNIQUE constraint σε (match_date, home_team_id, away_team_id) ώστε να μην καταχωρούνται διπλά. Τα home_team_id, away_team_id και το προαιρετικό country_id είναι FOREIGN KEY προς countries(country_id). Οι σκόρ (home_score, away_score) είναι NOT NULL INT, ενώ το neutral BOOLEAN υποδεικνύει ουδέτερο γήπεδο. Indexes σε match_date, home_team_id και away_team_id διευκολύνουν temporal queries και αναζητήσεις κατά ομάδα.

3) **goals**
   Ο πίνακας goals καταγράφει κάθε γκολ: goal_id (SERIAL) PK, match_id και team_id είναι NOT NULL FOREIGN KEYS σε matches και countries. Το scorer VARCHAR(255) μπορεί να είναι NULL αν δεν είναι γνωστό όνομα. Το minute INT δείχνει το λεπτό του αγώνα· τα flags own_goal και penalty είναι BOOLEAN με default FALSE. Indexes σε match_id, team_id και scorer επιταχύνουν στατιστικές αναλύσεις και αναζητήσεις ανά παίχτη.

4) **penalty_shootouts**
   Για αγώνες που κρίνονται στα πέναλτι, ο πίνακας penalty_shootouts έχει shootout_id (SERIAL) PK και μοναδικό FOREIGN KEY match_id σε matches, εξασφαλίζοντας ένα record ανά αγώνα. Το winner_id (NOT NULL) αναφέρεται σε countries, ενώ το first_shooter_id (nullable FK) αναγράφει ποια ομάδα εκτέλεσε πρώτη. Δεν υπάρχουν επιπλέον indexes, καθώς το μοναδικό match_id καλύπτει την πρόσβαση.

5) **former_names**
   Το ιστορικό των ονομάτων των χωρών κρατιέται εδώ: former_name_id

(SERIAL) PK, country_id NOT NULL FK σε countries. Το former_name VARCHAR(255) είναι υποχρεωτικό, και τα start_date/end_date (DATE) δείχνουν το διάστημα χρήσης. Δεν υπάρχουν επιπλέον unique constraints, γιατί μια χώρα μπορεί να αλλάξει όνομα πολλές φορές.

Οπτικοποίηση αλληλεξαρτήσεων πινάκων (με Foreign Keys) από το chartDB

Backend

Views

Μέθοδος	Route	Περιγραφή	Κύρια Query Params
GET	/api	Επιστρέφει μήνυμα καλωσορίσματος	—
GET	/api/match-details	Λίστα αγώνων με enriched πεδία (ομάδες, σκορ, τοποθεσία, έτος)	tournament, tournament_year, home_team, away_team, match_city, match_country, startYear, endYear, limit, offset, sort
GET	/api/goal-details	Λεπτομέρειες γκολ (παίκτης, ομάδα, λεπτό, τύπος γκολ)	tournament, tournament_year, scorer_name, scoring_team, team_conceded, is_penalty, is_own_goal, startYear, endYear, limit, offset, sort
GET	/api/tournament-summary	Στατιστικά ανά διοργάνωση & έτος (αγώνες, γκολ, αποφασιστικοί/ισοπαλίες)	tournament, year, startYear, endYear, limit, offset, sort
GET	/api/country-performance	Βασικά επιδόσεις κάθε χώρας (αγώνες, νίκες, ισοπαλίες, γκολ)	country_name, limit, offset, sort
GET	/api/team-tournament-performance	Απόδοση ομάδας ανά διοργάνωση & έτος	tournament, tournament_year, team_name, startYear, endYear, limit, offset, sort
GET	/api/head-to-head	Head-to-head στατιστικά δύο ομάδων	**Υποχρεωτικά**: team1_name, team2_name Προαιρετικά: startYear, endYear, sort
GET	/api/goal-timing	Κατανομή γκολ ανά χρονικά segments	tournament, startYear, endYear
GET	/api/country-profiles	Ολοκληρωμένο προφίλ χώρας (activity span, performance, home/away stats)	country_name, continent, region_name, sub_region_name, developed_or_developing,

Μέθοδος	Route	Περιγραφή	Κύρια Query Params
			startYear, endYear, limit, offset, sort
GET	/api/yearly-summary	Συνολική δραστηριότητα αγώνων ανά έτος & ήπειρο	year, startYear, endYear, limit, offset, sort
GET	/api/scorer-summary	Στατιστικά παικτών (σύνολο γκολ, πρώτος/τελευταίος χρόνος, max γκολ/αγώνα)	scorer_name, startYear, endYear, limit, offset, sort

Dynamic Queries

Μέθοδος	Route	Περιγραφή	Κύρια Query Params / Path Variables
GET	**/api/player-goals**	**Αναζήτηση γκολ παικτών με φίλτρα σε όνομα, χρονιές, διοργάνωση**	**scorerName, startYear, endYear, tournament, limit, offset, sort**
GET	**/api/match-list**	**Finder αγώνων με φίλτρα home/away/team/τουρνουά/πόλη/χώρα**	**homeTeam, awayTeam, team, tournament, startYear, endYear, city, country, limit, offset, sort**
GET	**/api/player-goal-timeline**	**Timeline γκολ παίκτη ανά έτος**	**scorerName (απαιτείται), startYear, endYear**
GET	**/api/country-wdl-timeline**	**Timeline νικών/ισοπαλιών/ηττών χώρας ανά έτος**	**countryName (απαιτείται), startYear, endYear**
GET	**/api/top-countries/:metric**	**Top-10 χώρες κατά metric (matches, goals, wins, draws, losses, win_ratio)**	**Path: metric Query: startYear, endYear, continent,**

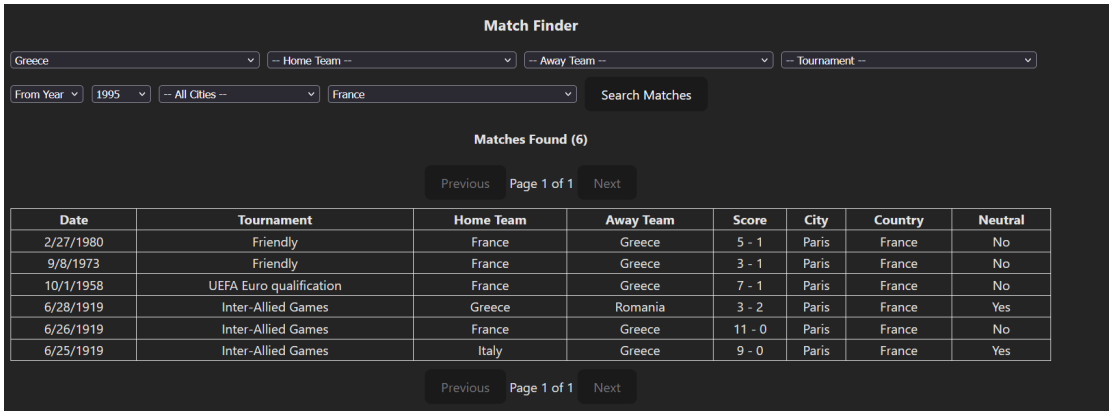
Μέθοδος	Route	Περιγραφή	Κύρια Query Params / Path Variables
			region_name, sub_region_name
GET	/api/country-activity	Activity summary (πρώτο/τελευταίο έτος, distinct έτη) για χώρα	countryName (απαιτείται)
GET	/api/distinct-tournaments	Λίστα όλων των distinct ονομάτων τουρνουά	—
GET	/api/distinct-countries	Λίστα χωρών (εξαιρούνται virtual με status='Unrecognized')	—
GET	/api/distinct-scorers	Λίστα distinct scorer, με optional φίλτρο με q	q
GET	/api/distinct-cities	Λίστα distinct πόλεων	—
GET	/api/distinct-match-years	Λίστα distinct ετών αγώνων	—
GET	/api/distinct-scoring-years	Λίστα distinct ετών σκοραρίσματος	—
GET	/api/distinct-active-years	Λίστα distinct ετών δραστηριότητας (country_profile)	—
GET	/api/distinct-continents	Λίστα distinct ηπείρων	—

Frontend

Component	Σκοπός	Λειτουργία (Functionality)
CountryActivity	Εμφάνιση περιόδου δραστηριότητας χώρας	Φορτώνει λίστα χωρών, επιτρέπει επιλογή χώρας και καλεί /country-activity; εμφανίζει first_year_active, last_year_active, distinct_years_played.
CountryProfileChart	Οπτικοποίηση συνολικής επίδοσης χώρας	Φορτώνει φίλτρα (χώρα, έτη), sort & limit, καλεί /country-profiles, σχεδιάζει bar chart με D3 για το επιλεγμένο metric και εμφανίζει λεπτομέρειες σε πίνακα.
CountryWdlChart	Timeline νικών/ισοπαλιών/ηττών για χώρα	Φορτώνει φίλτρα (χώρα, έτη), καλεί /country-wdl-timeline, συμπληρώνει κενά έτη με μηδενικά, σχεδιάζει γραμμικά διαγράμματα (wins, draws, losses) και εμφανίζει πίνακα με ετήσια στατιστικά.
GlobalTopStats	Top-10 στατιστικά global ανά ήπειρο	Φορτώνει λίστα ηπείρων, για κάθε metric (matches, wins, κλπ.) καλεί /top-countries/{metric}, εμφανίζει 10 πρώτες χώρες σε πίνακες side-by-side.
GoalTimingChart	Κατανομή γκολ ανά χρονικά segments	Φορτώνει φίλτρα (τουρνουά, έτη), καλεί /goal-timing, ταξινομεί segments, σχεδιάζει bar chart με D3 (time_segment vs count).
MatchFinder	Αναζήτηση αγώνων με πολλαπλά φίλτρα	Φορτώνει λίστες (χώρες, τουρνουά, πόλεις, έτη), δέχεται φόρμα φίλτρων, καλεί /match-list με pagination, εμφανίζει πίνακα αποτελεσμάτων και κουμπιά πλοήγησης.
PlayerAutocompleteSearch	Autocomplete για παίκτες	Φορτώνει όλους τους scorers, φιλτράρει με debounce σε text input, εμφανίζει λίστα suggestions, επιτρέπει επιλογή scorer για χρήση αλλού.

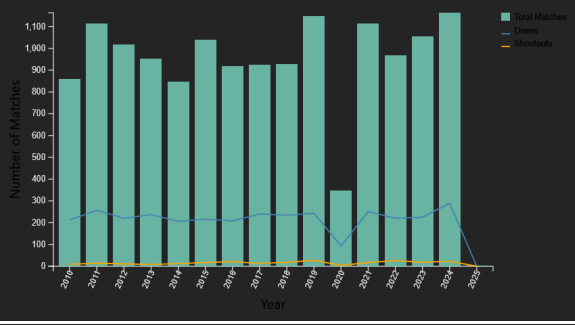
Component	Σκοπός	Λειτουργία (Functionality)
PlayerGoalSearch	Εύρεση γκολ παίκτη	Φορτώνει λίστες (scorers, τουρνουά), δέχεται φίλτρα (παίκτης, έτη, τουρνουά), καλεί /player-goals με pagination, εμφανίζει πίνακα γκολ και κουμπιά πλοήγησης.
PlayerProfile	Προφίλ παίκτη & timeline γκολ	Φορτώνει λίστες (scorers, έτη), καλεί παράλληλα /player-goal-timeline και /scorer-summary, σχεδιάζει line chart για timeline και εμφανίζει summary card με συνολικά goals & first/last year.
ScorerSummaryChart	Οπτικοποίηση summary scorers	Φορτώνει φίλτρα (scorer, έτη), sort & limit, καλεί /scorer-summary, σχεδιάζει bar chart με D3 για το επιλεγμένο metric (total_goals, κλπ.) και εμφανίζει λεπτομέρειες σε πίνακα.
YearlySummaryChart	Γραφική & συνοπτική παρουσίαση δραστηριότητας ανά έτος/ήπειρο	Φορτώνει φίλτρα (έτη, ήπειρο), καλεί /yearly-summary, συγκεντρώνει raw results, κατασκευάζει συνδυασμένα totals με group-by έτος, σχεδιάζει bar+line chart (total matches, draws, shootouts) και εμφανίζει πίνακα.

6. Snapshots



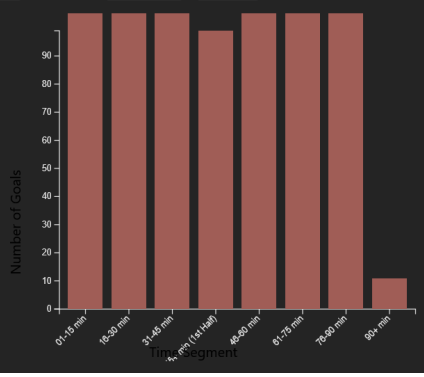
Yearly Match Summary (by Match Location Continent)

// Labels = DropdownsYear Range: 2010 - To Year Continent: Europe



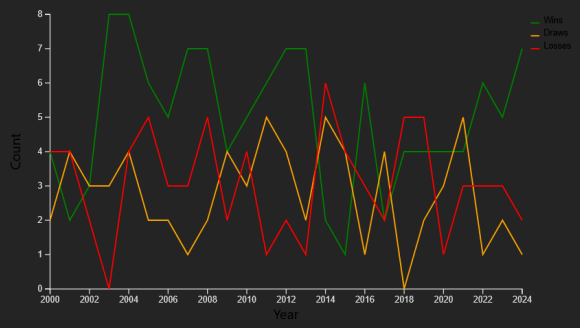
Goal Timing Distribution

Tournament: UEFA Euro Year Range: 2017 - To Year



Country Win/Draw/Loss Timeline

Country: Greece Year Range: 2000 - To Year



7. Outro

Συμπερασματικά, η παρούσα λύση καλύπτει απόλυτα τις ανάγκες για αξιόπιστη ανάλυση ποδοσφαιρικών αγώνων, μετά το φιλτράρισμα των CSV, την σωστή αποθήκευσή τους και τα views σε επίπεδο βάσης, αλλά και την ενδεχόμενη εξυπηρέτηση πολλών χρηστών ταυτόχρονα λόγω της ασύγχρονης φύσης του node. Παράλληλα, ο ξεκάθαρος διαχωρισμός ETL, DB, API, UI διευκολύνει τη συντήρηση και επιτρέπει την ανεξάρτητη ανάπτυξη κάθε υποσυστήματος.

Μελλοντικά, θα μπορούσα να το τοποθετήσω σε containers και να αναπτύξω μηχανισμούς authentication/authorization (JWT, OAuth2) και rate limiting για προστασία των endpoints ώστε να το μοιραστώ δημόσια ή ακόμη να αξιοποιήσω προγνωστικά μοντέλα για προβλέψεις